REMARKS/ARGUMENTS

Applicant respectfully requests reconsideration of this application in view of the following remarks.

Claim 1 Rejection under 35 U.S.C. § 103(a) – Noy in view of Jacobs

Applicant's claim 1 recites:

1. A computer network architecture comprising:

(a) a first layer including a transmission control protocol connection;

(b) a second layer including a hyper text transfer protocol connection built upon the first layer;

(c) a first tunneling layer including a first tunneling protocol built upon the second layer to tunnel a message through the hyper text transfer protocol connection; and

(d) a multiplexing layer to multiplex a plurality of messages for transmission through the first tunneling layer.

[Bolded markers for (a), (b), (c), (d) added for discussion below.]

With respect to **(c)**:

The Office states in paragraph 3 page 2:

Noy discloses ...

a first tunneling layer including a first tunneling protocol built upon the second layer to tunnel a message through the hyper text transfer protocol connection; [Noy, Multi HTTP request/response channels tunneled over individual TCP connections, TCP based and tunneled, multi-part channels, col 8 lines 45-60].

Noy at the cited section col 8 lines 45-60 states:

Preference of Client-side Execution Environments

Different client-side code execution environments have different features in terms of the type of communication channels the execution environment can establish. For example, a typical Java virtual machine execution environment has the built-in capability to establish TCP-based communication channels, as well as multi-request/response communication channels *tunneled over individual TCP connections* (e.g., multiple HTTP requests with channel-identifying or session-identifying cookies). Dynamic scripting environments typically only have the built-in ability to create communication channels tunneled over multiple request/response transactions. Networked ActiveX execution environments typically also provide the ability to make TCP-based and tunneled, multi-part communication channels.

(Emphases added.)

Firstly, Noy is describing "multi-request/response communication channels tunneled over individual TCP connections." (Emphases added.) In contrast Applicant's claim 1 recites "to tunnel a message through the hyper text transfer protocol connection." (Emphasis added.) Tunneling through individual TCP connections (Noy) is not the same as tunneling through HTTP (Applicant) because individual TCP connections (Noy) are not the same as a HTTP connection (Applicant).

Secondly, Noy when referencing "multiple HTTP requests with channel-identifying or session-identifying cookies" is still referring to using the individual TCP connections. Multiple HTTP requests with channel IDs tunneling through individual TCP connections (Noy) is not the same as tunneling through the HTTP connection (Applicant) because HTTP tunneling through individual TCP connections (Noy) is different than tunneling through HTTP.

With respect to **(d)**:

The Office states in paragraph 3 page 3:

"Jacob disclosed the messages can be transferring using different protocols: TCP/IP, HTTP tunneling and multiplexes the threads between socket reading and request execution, col 8 lines 40-62]"

Jacobs at the cited section col 8 lines 40-62 states:

In particular, server 302, server 303, and client 304 have kernels 302b, 303b, and 304b, respectively. In particular, in order for two JVMs to interact, whether they are clients or servers, each JVM constructs an RJVM representing the other. Messages are sent from the upper layer on one side, through a corresponding RJVM, across the communication medium, through the peer RJVM, and delivered to the upper layer on the other side. In various embodiments, messages can be transferred using a variety of different protocols, including, but not limited to, Transmission Control Protocol/Internet Protocol ("TCP/IP"), Secure Sockets Layer ("SSL"), Hypertext Transport Protocol ("HTTP") tunneling, and Internet InterORB Protocol ("IIOP") tunneling, and combinations thereof. The RJVMs and socket managers create and maintain the sockets underlying these protocols and share them between all objects in the upper layers. A socket is a logical location representing a terminal between processing devices in a distributed processing system. The kernel maintains a pool of execute threads and thread manager software component 364 multiplexes the threads between socket reading and request execution. A thread is a sequence of executing program code segments or functions.

(Emphases added.)

Firstly: Jacobs is discussing sockets and multiplexing executing code segments between socket reading and request execution. Multiplexing executing code segments (Jacobs) is not the same as a multiplexing layer to multiplex a plurality of messages for transmission through the first tunneling layer (Applicant), because executing code (Jacobs) is not the same as multiplexing messages (Applicant).

Overall for Claim 1

First, Noy does not show tunneling a message through the HTTP. Second, Jacobs does not cure this defect. Additionally neither Noy nor Jacobs shows multiplexing a plurality of messages for transmission through the HTTP layer. Thus, the combination of Noy in view of Jacobs does not show "a first tunneling layer including a first tunneling protocol built upon the second layer to tunnel a message through the hyper text transfer protocol connection; and a multiplexing layer to multiplex a plurality of messages for transmission through the first tunneling layer" which Applicant has claimed.

For the above reasons, Applicant submits that Applicant's claim 1 is not obvious in view of Noy in view of Jacobs. Applicant therefore requests that this rejection be withdrawn and that claim 1 and all claims dependent on claim 1 be allowed.

Claim 2 Rejection

Claim 2 is dependent on claim 1 and Applicant submits that for all the reasons in the claim 1 discussion that Applicant's claim 2 is not obvious in view of Noy in view of Jacobs. Additionally, Noy at the cited lines (col 8, lines 45-60) discloses TCP tunneling which is not the same as Applicant's claim 2 "wherein the first tunneling protocol opens the hyper text transfer protocol connection between a server and a client ."

For the above reasons, Applicant submits that Applicant's claim 2 is not obvious in view of Noy in view of Jacobs. Applicant therefore requests that this rejection be withdrawn and that claim 2 and all claims dependent on claim 2 be allowed.

Claims 10-11, and 19-20 Rejection under 35 U.S.C. § 103(a) – Noy in view of Jacobs

The Office at 5. states:

> 5. Claims 10-11,19-20 contain the similar limitations set forth of apparatus claims 1- 2. Therefore, claims 10-11,19-20 are rejected for the similar rationale set forth in claims 1-2.

For the same reasons as noted above in the claim 1 and claim 2 discussion, Applicant submits Applicant's claims 10-11, and 19-20 are not obvious in view of Noy in view of Jacobs. Applicant therefore requests that these rejections be withdrawn and that claims 10-11, and 19-20 and all claims dependent on claims 10-11, and 19-20 be allowed..

Claim 3 Rejection under 35 U.S.C. § 103(a) – Noy in view of Jacobs further in view of Pujare

Claim 3 is dependent on claim 1 and Applicant submits that for all the reasons in the claim 1 discussion that Applicant's claim 3 is not obvious in view of Noy in view of Jacobs.

Additionally, the Office states at 6:

> 6. As per claim 3, Noy discloses a first layer (TCP) tunneling through a second layer (HTTP). However Noy does not detail "a second tunneling layer including a second tunneling protocol built upon the first layer to tunnel a message through the TCP connection."

In the same endeavor, Pujare discloses a conventionally coded application conversion system wherein the component that does the network spoofinq is the TCP - HTTP converter includinq TCP packets tunnel through HTTP on one side and do exactly the opposite on the other [Pujare, 0156]

Therefore it would have been obvious to an ordinary skill in the art at the time the invention was made to incorporate the Pujare teaching into the Noy's apparatus in order to utilize the channel construction mechanism. Doing so would provide a conventionally coded application conversion for streamed delivery and execution which does not require the recompiled [Pujare,0014].

(Emphasis added.)

At the cited reference Pujare states:

[0156] Referring to FIGS. 1, 4, 5a, 5b, and 6a, the component that does the Network Spoofing is the TCP to HTTP converter 503, 507. The basic idea is to take TCP packets and tunnel them through HTTP on one side and do exactly the opposite on the other.

(Emphasis added.)

Taking TCP packets and tunneling them through HTTP on one side and the opposite on the other (Pujare) is not the same as a second tunneling layer including a second tunneling protocol built upon the first layer to tunnel a message through the TCP connection (Applicant).

Applicant therefore requests that this rejection be withdrawn and that claim 3 and all claims dependent on claim 3 be allowed.

Claim 4 Rejection

Claim 4 is dependent on claim 3 which is dependent on claim 1 and Applicant submits that for all the reasons in the claim 1 and claim 3 discussions above that Applicant's claim 4 is not obvious in view of Noy in view of Jacobs further in view of Pujare.

Applicant therefore requests that this rejection be withdrawn and that claim 4 and all claims dependent on claim 4 be allowed.

Claim 5 Rejection

Claim 5 is dependent on claim 4 is dependent on claim 3 which is dependent on claim 1 and Applicant submits that for all the reasons in the claim 1, claim 3, and claim 4 discussions above that Applicant's claim 5 is not obvious in view of Noy in view of Jacobs further in view of Pujare.

Applicant therefore requests that this rejection be withdrawn and that claim 5 be allowed.

Claim 6 Rejection

Claim 6 is dependent on claim 1 and Applicant submits that for all the reasons in the claim 1 discussion that Applicant's claim 6 is not obvious in view of Noy in view of Jacobs further in view of Pujare. Applicant therefore requests that this rejection be withdrawn and that claim 6 be allowed.

Claim 7 Rejection

Claim 7 is dependent on claim 1 and Applicant submits that for all the reasons in the claim 1 discussion that Applicant's claim 7 is not obvious in view of Noy in view of Jacobs further in view of Pujare. Applicant therefore requests that this rejection be withdrawn and that claim 7 and all claims dependent on claim 7 be allowed.

Claim 8 Rejection

Claim 8 is dependent on claim 7 which is dependent on claim 1 and Applicant submits that for all the reasons in the claim 1 discussion and the claim 7 discussion above that Applicant's claim 8 is not obvious in view of Noy in view of Jacobs further in view of Pujare. Applicant therefore requests that this rejection be withdrawn and that that claim 8 be allowed.

Claim 9 Rejection

Claim 9 is dependent on claim 7 which is dependent on claim 1 and Applicant submits that for all the reasons in the claim 1 discussion and the claim 7 discussion above that Applicant's claim 9 is not obvious in view of Noy in view of Jacobs further in view of Pujare.

Applicant therefore requests that this rejection be withdrawn and that claim 9 be allowed.

Claims 12-18 and 21-27

Applicant submits that for the same reasons explained above in the various claims discussions, that claims 12-18 and 21-27 are not obvious in view of Noy in view of Jacobs further in view of Pujare. Applicant therefore requests that claims 12-18 and 21-27 be allowed.

CONCLUSION

Applicant respectfully submits that all claims are in condition for allowance, and requests allowance of all claims.

The Examiner is invited to call Alan Heimlich at 408 253-3860 if there remains any issue with allowance. Email communication is authorized.

Respectfully submitted,

Heimlich Law

Digitally signed by Alan Heimlich
DN: CN = Alan Heimlich, C = US, O = Heimlich Law

Alan Heimlich / Reg 48808

Attorney for Applicant(s)

02/13/2006

Date

Customer No. 40418

5952 Dial Way
San Jose, CA 95129

Tel: 408 253-3860
Eml: alanheimlich@heimlichlaw.com